



Institut Supérieur d'Enseignement Professionnel – Diamniadio

N°: DISEP\_ TIC \_P1 \_ Groupe 17

PROJET DE FIN DE FORMATION POUR L'OBTENTION DU DISEP  
DIPLOME DES INSTITUTS SUPERIEURS D'ENSEIGNEMENT PROFESSIONNELS

Thème :

# Réalisation d'une application de gestion d'établissement scolaire

Soutenu le : 06/ 12 / 2021

Présenté par :

1. Omar THIOMBANE
2. Coumba SOW
3. Fatimata WANE
4. Khoudia NIANG
5. Adama DIEYE
6. Coumba NDIAYE

Encadreur : M. Cheikh Hamet Tidiane Cherif NDIAYE

Année académique : 2020 – 2021

Thème :

# Réalisation d'une application de gestion d'établissement scolaire

---

## *Épître*

*« Les jeunes doivent se prendre en main et s'assurer qu'ils reçoivent le meilleur enseignement possible, afin de pouvoir bien nous représenter à l'avenir. »*

*Nelson Mandela*

*« Si vous vous endormez en pensant que quelque chose est impossible, vous risquez d'être réveillé par le bruit qu'un autre fera en l'accomplissant. »*

*Jean de La Bruyère*

## DEDICACES

Nous dédions ce travail

A nos très chers parents.

A nos chers frères et sœurs.

A tous les membre de la famille et à nos amis.

A tous les apprenants de l'ISEPDD.

Et à tous les gens qui nous ont aidé et soutenu tout au long de ce projet.

Pour tous vos sacrifices, nul mot ne saura exprimer notre amour envers vous. Que Dieu vous protège et vous accorde une longue vie.

## REMERCIEMENT

Tout d'abord, nous remercions Dieu de nous avoir donné la volonté de terminer ce document.

Nous tenons également à remercier et à exprimer notre gratitude aux personnes suivantes pour leur engagement et leur soutien dans la réalisation de ce projet :

- M. Amar DIOP, Directeur des études à l'isep de Diamniadio, pour la confiance qu'il nous a accordée dès le début du projet et pour nous avoir permis de le mettre en œuvre.
- M. Cheikh Hamet Tidiane Cherif NDIAYE, Formateur, pour sa coopération professionnelle au cours de cette expérience et pour nous avoir apporté une partie de son savoir-faire et de son expérience professionnelle.
- Mr. Serigne Mbacké KANDJI, Responsable projet, pour ses conseils éclairés, sa pertinence, sa disponibilité.
- Pr Omar NIAN, Ancien Directeur Général de l'isep de Diamniadio, pour son encouragement continu et ses conseils avisés.

Nous remercions également les formateurs de l'ISEP de Diamniadio pour leur gentillesse, leurs précieux conseils, leur disponibilité et leur soutien.

Tous nos remerciements vont à nos parents, frères et sœurs pour leur dévouement et leur soutien durant ces deux années d'études.

Enfin, nous remercions tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.

## Sigle et abréviation

- **ISEPDD** : Institut Supérieur d'Enseignement Professionnel de Damniadio
- **UML** : Langage de modélisation unifié
- **SGBD** : système de gestion de base de données
- **HTML** : HyperText Markup Language
- **CSS** : Cascading Style Sheets,
- **MVT** : Modèle Vue Template
- **POO** : programmation orienté objet
- **MVC** : Modèle Vue Contrôleur
- **CSS** : Cascading Style Sheets, feuilles de style en cascade en français
- **MPD** : Modèle Physique de Données

## Liste des figures

- Figure1 : Les vues du langage UML
- Figure2 : les différents diagrammes UML
- Figure3 : Interactions entre le modèle, la vue et le template
- Figure4 : Diagramme de cas d'utilisation générale
- Figure5 : Diagramme de cas d'utilisation «s'authentifier »
- Figure6 : Diagramme de cas d'utilisation «paramétrage »
- Figure7 : Modèles Physiques des Données
- Figure8 : Interface d'authentification
- Figure9 : Interface 2 « Administrateur »
- Figure10 : Interface « Éditer profil »
- Figure11 : Interface « Ajouter professeur »
- Figure12 : Interface « Gestion professeur »
- Figure13 : Interface « Envoyer notification au professeur »
- Figure14 : Interface « Message de rétroaction des professeurs»
- Figure15 : Interface « Demande des professeurs »
- Figure16 : Interface « Ajouter Apprenants »
- Figure17 : Interface « Gestion Apprenants »
- Figure18 : Interface « Voir Résultat »
- Figure19 : Interface « Envoyer notification à l'élève »
- Figure20 : Interface « Message de rétroaction des élèves »
- Figure21 : Interface Lister demande élèves »
- Figure22 : Interface « Ajouter classes »
- Figure23 : Interface « Gestion classe »
- Figure24 : Interface « Ajouter Matière »
- Figure25 : Interface « Gestion Matière »
- Figure26 : Interface « Ajouter Session »
- Figure27 : Interface « Gestion Session »
- Figure28 : Interface « Afficher Présence »
- Figure29 : Interface « Accueil professeur »

Figure30 : Interface « Vue/Mise à jour » professeur

Figure31 : Interface « Téléchargement des résultats »

Figure32 : Interface « Éditer résultats »

Figure33 : Interface « Demande d'autorisation d'absence »

Figure34 : Interface « Mise à jour absence »

Figure35 : Interface « Notification »

Figure36 : Interface « Demander un congé »

Figure37 : Interface « Ajouter commentaires »

Figure38 : Interface « Accueil élève »

Figure39 : Interface « Notes élèves »

Figure40 : Interface « Éditer Profil élève »

Figure41 : Interface « Présence »

Figure42 : Interface « Demande autorisation d'absence élève »

Figure43 : Interface « Commentaire élève »

## Résumé

Dans le cadre de l'obtention du Diplôme de Technicien Supérieur en Développement Backend de l'Institut Supérieur d'Enseignement Professionnel de Diamniadio, il nous a été demandé de réaliser un programme de fin de formation pour clore notre formation de deux ans. C'est ainsi que nous avons la possibilité d'approfondir nos connaissances théoriques et pratiques en concevant et en construisant un logiciel de gestion scolaire complet. Cette application est conçue pour réduire les tâches chronologiques et répétitives, tout en générant plus d'interactions entre les écoles, les enseignants, les élèves et les parents d'élèves pour faciliter les tâches aux établissements secondaires. Son objectif principal est donc de redynamiser et d'organiser l'enseignement, et surtout, de gérer les programmes scolaires des élèves. Afin d'atteindre ces objectifs, nous avons créé une application Web basée sur la modélisation du langage UML. Le langage de programmation utilisé est Python et le système de gestion de base de données (SGBD) est MySQL. Ce travail a été réalisé sous la direction de notre formateur, M. Cheikh Hamet Tidiane Cherif NDIAYE.



## Abstract

As part of the Diploma of Superior Technician in Backend Development of the ISEP of Diamniadio, we were asked to carry out an end of training program to complete our two-year training. Thus, we have the opportunity to deepen our theoretical and practical knowledge by designing and building a complete school management software. This application is designed to reduce chronological and repetitive tasks, while generating more interactions between schools, teachers, students and parents to facilitate the tasks of secondary schools. Its main objective is therefore to revitalize and organize teaching, and above all, to manage the students' school programs. In order to achieve these objectives, we have created a web application based on UML modeling. The programming language used is Python and the database management system (DBMS) is MySQL. This work was carried out under the direction of our trainer, Mr. Cheikh Hamet Tidiane Cherif NDIAYE.

## SOMMAIRE

<b>INTRODUCTION .....</b>	<b>11</b>
<b>1.CONSTAT DE DÉPART .....</b>	<b>12</b>
1.1. Contexte général du projet.....	12
1.2. Présentation général du projet .....	13
1.3. Objectif du projet.....	13
<b>2.CADRE THÉORIQUE .....</b>	
<b>3.ETUDE FONCTIONNEL ET TECHNIQUE.....</b>	<b>15</b>
3.1. Spécification des besoins.....	15
3.1.1. Spécification des besoins fonctionnels .....	15
3.1.2. Spécification des besoins non fonctionnels .....	16
3.2. Choix méthodologiques.....	17
3.2.1. Le langage UML.....	18
3.2.2. La technologie orienté objet .....	20
3.2.3. Le modèle MVT .....	21
3.2.4. Le Framework Django .....	23
3.3. Les langages utilisés .....	24
3.3.1. HTML .....	24
3.3.2. CSS .....	24
3.3.3. Python .....	24
3.3.4. JS.....	24
<b>4. ANALYSE ET CONCEPTION .....</b>	<b>26</b>
4.1. Modélisation en langage UML .....	26
4.1.1. Identification des acteurs .....	26
4.1.2 Le diagramme de cas d'utilisation .....	26
4.2. Mise en place de la base de donnée.....	29
4.2.1. Modèle Physique de Données.....	29

<b>5. IMPLÉMENTATION ET RÉSULTAT.....</b>	<b>31</b>
5.1. Plateforme de développement .....	31
5.1.1. SGBDR MySQL.....	31
5.1.2. Le langage de développement Python .....	32
5.2. Les interfaces de l'application.....	32
<b>6. CONCLUSION.....</b>	<b>52</b>
<b>7. PERSPECTIVES.....</b>	<b>53</b>
<b>WEBOGRAPHIE.....</b>	<b>54</b>

## INTRODUCTION

Aujourd'hui, le domaine informatique représente la révolution la plus nécessaire et la plus innovante de la vie humaine au 21<sup>e</sup> siècle.

En fait, l'informatique est loin d'être une mode éphémère ou une tendance dépassée, mais apporte une variété d'applications à notre mode de vie. Cette stratégie rend tant de services à l'entreprise ou au service administratif et aux employés qu'aucun domaine n'est étranger.

Cependant, outre l'utilisation personnelle des ordinateurs, le plus important est la communication informatique, qui permet de changer radicalement les méthodes de travail. Jusqu'à présent, nous avons assisté à l'essor du développement informatique. Ce nouveau développement offre aux utilisateurs de nouveaux outils de travail qui leur permettent d'augmenter leur rentabilité et leur productivité.

C'est dans ce cadre que s'inscrit également notre projet de fin de formation appelé *Souniou School* qui vise à informatiser le système de gestion des élèves, des professeurs et du personnel, et de leurs utilisateurs.

Dans ce rapport, nous définissons d'abord le contexte général du projet, c'est-à-dire l'introduction globale du projet et ses objectifs.

Deuxièmement, nous définissons les opérations et les diverses fonctions fournies par l'application.

Ensuite, nous nous intéressons à l'utilisation de diagrammes de classes UML et de cas d'utilisation pour concevoir différents modules de notre projet.

Ensuite, nous regardons la recherche technique et la mise en œuvre de notre projet. En fait, nous allons d'abord décrire l'architecture technique, les outils et les frameworks utilisés pour le développement. Ensuite, nous expliquerons le processus de développement et illustrerons certaines des fonctions fournies par notre application avec quelques captures d'écran.

Enfin, une conclusion qui résume les travaux réalisés dans le cadre du projet de fin de formation et propose des pistes d'amélioration susceptibles d'améliorer encore les outils développés.

## 1. CONSTAT DE DÉPART

### 1.1.Contexte général du projet

En effet, la nécessité d'usage de l'outil informatique étant dans tous les domaines presque, dans le monde actuel, c'est ainsi que nous étudiants de l'Institut Supérieure d'enseignements professionnels de Diamniadio(ISEPD) ayant dans ses objectifs de la promotion des travaux de qualité en informatique notamment, l'automatisation de la gestion des affaires, se fixe comme soucis d'aider les institutions tant publiques que privées à bien gérer et planifier leur système de gestion, c'est-à-dire donner aux Top Managers des systèmes adéquats de la gestion de leurs établissements en vue de gagner le temps, de conserver des données, de limiter des erreurs de calcul et pas mal d'autres raisons, chose qui ne nous laisse pas indifférents, du fait qu'on se heurte à certains problèmes dans les établissements, dont :

- Les erreurs de calculs,
- Le gaspillage du temps pendant les calculs,
- Perte des feuilles de devoir, ou de compositions
- Difficulté voire même l'impossibilité aux surveillants de pouvoir produire les rapports journaliers,
- Difficulté aux élèves de prouver leur paiement de frais scolaires.

De cela, nous avons compris que tous ces différents problèmes sont dus du fait que l'école gère manuellement sa perception des frais scolaires des élèves, c'est-à-dire qu'elle utilise des papiers en prenant tous les risques qui peuvent en découler,

C'est ainsi que nous nous sommes posée la question suivante :

-Que peut-on faire pour rendre rationnel le fonctionnement des établissements scolaires afin d'éviter les problèmes y afférents ?

Cette question constitue l'objet même du présent travail à laquelle nous allons chercher des solutions .

## 1.2.Présentation général du projet

Ce projet de fin de formation s'inscrit dans le cadre d'une solution pour la gestion des établissements scolaires. Cette solution doit être adaptée au système éducatif de nos lycées et collèges. La répartition des classes, des élèves, des matières, des départements, des filières, des spécialités, le calcul des moyennes et l'établissement des résultats se faisant de manière semi- automatique.

Nos établissements ne disposent d'aucune base de données permettant l'enregistrement et la consultation des résultats, statistiques de réussites, historiques des promotions de l'établissement. Les notes sont traitées manuellement, le traitement des relevés de notes prend assez de temps. Ces méthodes constituent un risque car un simple mélange de documents pourrait être fatal.

Une application étant définie comme un programme (ou un ensemble logiciel) directement utilisé par l'utilisateur pour réaliser une tâche, ou un ensemble de tâches élémentaires d'un même domaine ou formant un tout, sa conception dans notre cas consisterait à mettre en place une application web pour la gestion des écoles.

## 1.3.Objectifs du projet

L'objectif du projet est de développer une application web qui permet la gestion scolaire exhaustive, utile à l'architecture du système éducatif des établissements publics et privés du Sénégal. Cette application doit assurer la gestion automatisé de toutes les informations concernant les élèves, les enseignants et l'Administration.

Les principaux objectifs sont :

- Mobiliser parents d'élèves, communautés et équipes pédagogique pour le développement de l'éducation ;
- Élaborer et mettre en œuvre des activités d'amélioration de l'accès à l'éducation, de la qualité des enseignements apprentissages et de la gestion du système éducatif ;
- Digitaliser les activités administratives et pédagogique au niveau d'un établissement scolaire pour éviter les pertes de données en cas d'accident

Ce rapport décrit la démarche suivie, il est structuré de la manière suivante : après présentation de l'établissement une brève introduction du projet et la présentation des objectifs initiaux. Les outils de développement du logiciel sont aussi présenté et les différentes phases de sa réalisation.

Enfin une mise en perspective conclut le rapport.

En somme, nous avons présenté le contexte général du projet ainsi que ses objectifs ce qui nous a permis de recenser toutes les informations nécessaires et indispensables dans la réalisation de notre projet. Ces informations nous permettront de cerner les principaux objectifs et de choisir la technique à adopter, le contexte dans lequel nous réalisons ce projet et la solution retenue pour atteindre notre objectif. Nous pouvons dès lors passer à la partie suivante qui est l'étude technique et fonctionnelle de notre projet.

## 2. ETUDE FONCTIONNELLE ET TECHNIQUE

### 2.1. Spécification des besoins

La spécification de besoins constitue la phase de départ de toute application à développer dans laquelle nous allons identifier les différents besoins. Elle doit décrire sans ambiguïté le logiciel à développer. Elle est constituée d'un ensemble de documents et de modèles.

Toutes les personnes impliquées dans le projet doivent avoir accès à la spécification des besoins. Nous distinguons les besoins fonctionnels qui présentent les fonctionnalités attendues de notre application des besoins non fonctionnels qui quant à eux permettent d'éviter le développement d'une application non satisfaisante.

#### 2.1.1. Spécification des besoins fonctionnels

Les besoins fonctionnels ou besoin métiers représentent les actions que le système doit exécuter, il ne devient opérationnel que s'il les satisfait. Ils expriment une action que doit effectuer le système en réponse à une demande (sorties qui sont produites pour un ensemble donné d'entrées).

L'application que nous allons développer devra regrouper les fonctionnalités suivantes :

#### **A. Fonction des compte administrateur :**

- Seuls les administrateurs peuvent supprimer n'importe quel enregistrement
- Créer n'importe quel compte utilisateur
- Gérer les classes/sections des élèves
- Voir la feuille de note des élèves
- Créer, modifier et gérer les examens et les notes
- Créer, éditer et gérer des sujets
- Les avis sont visibles dans le calendrier du tableau de bord
- Modifier les paramètres système

#### **B. Fonction des comptes enseignant :**

- Gérer sa propre classe/section
- Gérer les dossiers d'examen pour ses propres sujets



- Gérer l'emploi du temps si affecté en tant que professeur de la classe
- Gérer son profil

**C. Fonction des comptes élèves :**

- Voir le profil de l'enseignant
- Voir ses propres sujets de classes
- Voir ses propres notes et horaire des cours
- Afficher le tableau d'affichage et les événements de l'école dans le calendrier
- Gérer son profil

**2.1.2. Spécification des besoins non fonctionnels**

Les besoins non fonctionnels sont des exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt ils identifient les contraintes internes et externes du système. Les principaux besoins non fonctionnels de notre application sont les suivants :

- ***Besoin de sécurité***

Il définissent les niveaux d'accès possibles au système pour les utilisateurs. Dans notre cas, cette application doit avoir un niveau de sécurité assez élevé, les comptes des utilisateurs devront être sécurisés par des mots de passe. Ces mots de passe seront individuels et devront respecter certaines conditions (la longueur du code; le code système, l'expiration de session, etc.). Les besoins de sécurité sont satisfaits par Django qui utilise un algorithme de cryptage évolué.

- ***Besoin de disponibilité***

Ils concernent le niveau de disponibilité qui doit être explicitement défini pour les applications critiques (Exemple : exigence de disponibilité 24h/ 24, 7j/ 7 sauf période de maintenance, à spécifier). Cette application devra fonctionner de manière efficace et ceux, sans défaillance. (Les utilisateurs pourront compter sur sa fiabilité. Ce besoin de disponibilité est satisfait grâce à l'hébergeur )

- ***Besoin de performance***

Ils décrivent les performances d'exécution du système, généralement en matière de temps de réponse (l'exemple d'une application Web), temps de chargement d'une page: le chargement d'une page Web dans le navigateur ne devrait pas prendre plus de 15 secondes en condition normale. Cette application devra répondre aux exigences des utilisateurs de façon optimale c'est-à-dire effectuer des opérations dans un laps de temps très court. (Ce besoin est garanti par le Framework. )

- ***Besoin de portabilité***

Cette application sera multiplateforme. Elle fonctionnera sur tous les systèmes d'exploitation et tout type de terminal puisqu'il s'agit d'une application web, elle sera disponible sur tout support où il existe un navigateur. Aussi, cette application sera responsive c'est-à-dire qu'elle s'adaptera à la taille de l'écran. Cette responsivité est assurée par bootstrap. On peut affirmer que ce besoin est déjà satisfait par la définition d'une application web.

- ***Une solution ouverte et évoluée***

L'application pourra être améliorée par l'ajout d'autres modules pour garantir sa souplesse, l'évolutivité et l'ouverture de la solution. On peut affirmer que notre application réponds au critère d'une solution ouverte et évoluée.

## 2.2.Choix méthodologique

Pour mener à bien notre projet, nos choix se sont portés sur les méthodologies et technologies suivantes :

- Le langage UML
- La technologie orienté objet
- Le modèle MVT
- Framework Django

### 2.2.1. Le langage UML

UML (Unified modeling Language) est un langage de modélisation unifié qui permet de modéliser une application logicielle d'une façon standard dans le cadre de conception orientée objet.

UML permet de couvrir le cycle de vie d'un logiciel depuis la spécification des besoins jusqu'au codage en offrant plusieurs moyens de description et de modélisation des acteurs et des utilisations du système, du comportement des objets, du flot de contrôles internes aux opérations, des composants d'implémentation et leurs relations, de la structure matérielle et de la distribution des objets et des composants indépendamment des techniques d'implémentation et peut-être mise à jour selon les besoins.

### **A. Les différentes vue du langage UML**

Une façon de mettre en œuvre UML est de considérer différentes vues qui peuvent se superposer pour collaborer à la définition du système. Ainsi on a :

- **La vue des cas d'utilisation :** c'est la description du modèle vu par les acteurs du système. Elle correspond aux besoins attendus par chaque acteur (c'est le QUOI et le QUI).
- **La vue logique :** c'est la définition du système vu de l'intérieur. Elle explique comment satisfaire les besoins des acteurs (c'est le COMMENT).
- **La vue d'implémentation :** cette vue définit les dépendances entre les modules.
- **La vue des processus :** c'est la vue temporelle et technique, qui met en œuvre les notions de tâches concurrentes, stimuli, contrôle, synchronisation, etc.
- **La vue de déploiement :** cette vue décrit la position géographique et l'architecture physique de chaque élément du système (c'est le OÙ).

Ci-dessous une figure qui représente ces différentes vues :

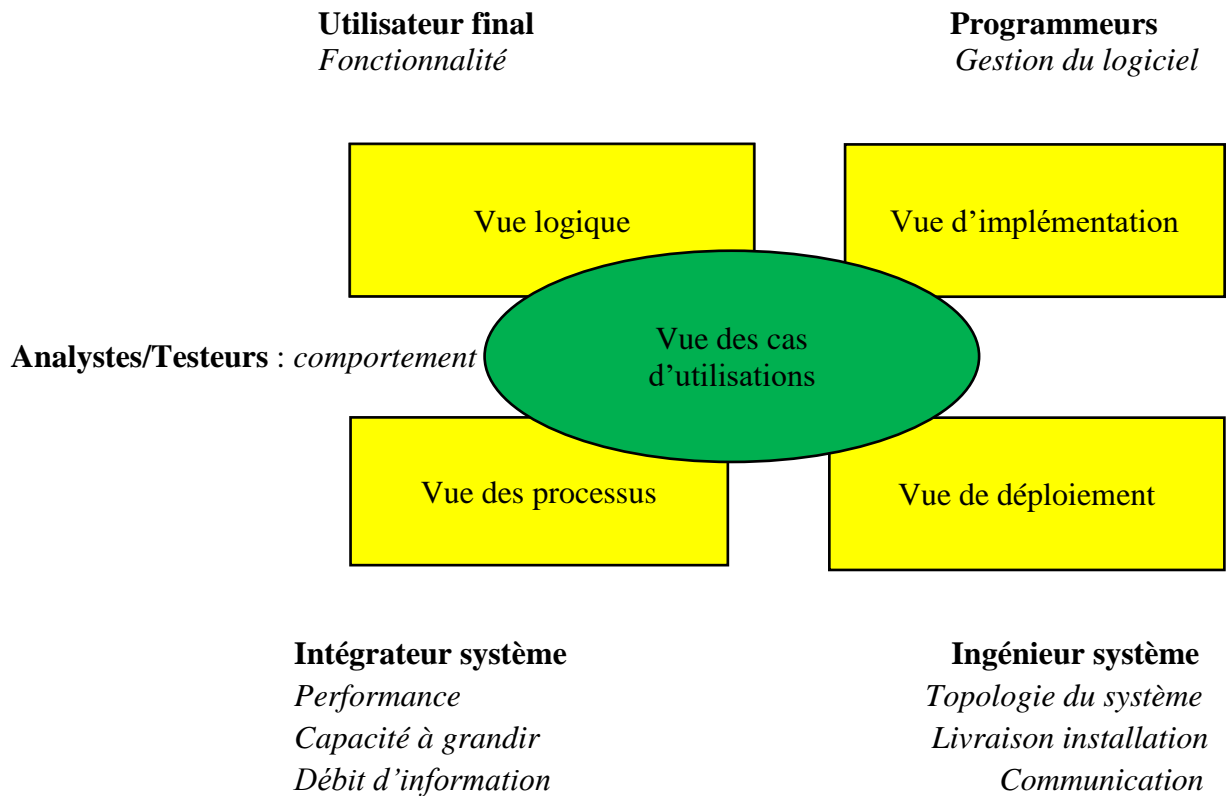


Figure1 : Les vues du langage UML

## B. Les principaux diagrammes UML

On distingue 9 principaux diagrammes UML qui sont dépendants hiérarchiquement et qui se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie. On distingue :

- **Le diagramme de cas d'utilisation** qui représente les fonctions du système du point de vue de l'utilisateur ;
- **Le diagramme d'objets** qui représente les objets et leurs relations, il montre comment le système est vu à l'instant t donné dans le temps;
- **Le diagramme de classes** qui représente la structure statique en termes de classes et de relations entre elles;
- **Le diagramme de composants** qui représente les composants physiques d'une application;
- **Le diagramme de déploiement** qui décrit le déploiement des composants sur les dispositifs matériels;

- **Le diagramme de collaboration** qui est une représentation spatiale des objets, des liens et des interactions ;
- **Le diagramme de séquence** qui représente temporellement les objets et leurs interactions;
- **Le diagramme d'état** qui décrit le comportement d'une classe d'objet en termes d'états ;
- **Le diagramme d'activités** montre les enchainements des activités d'un cas d'utilisation ou d'une opération.

Ces diagrammes sont regroupés dans la figure ci-après :

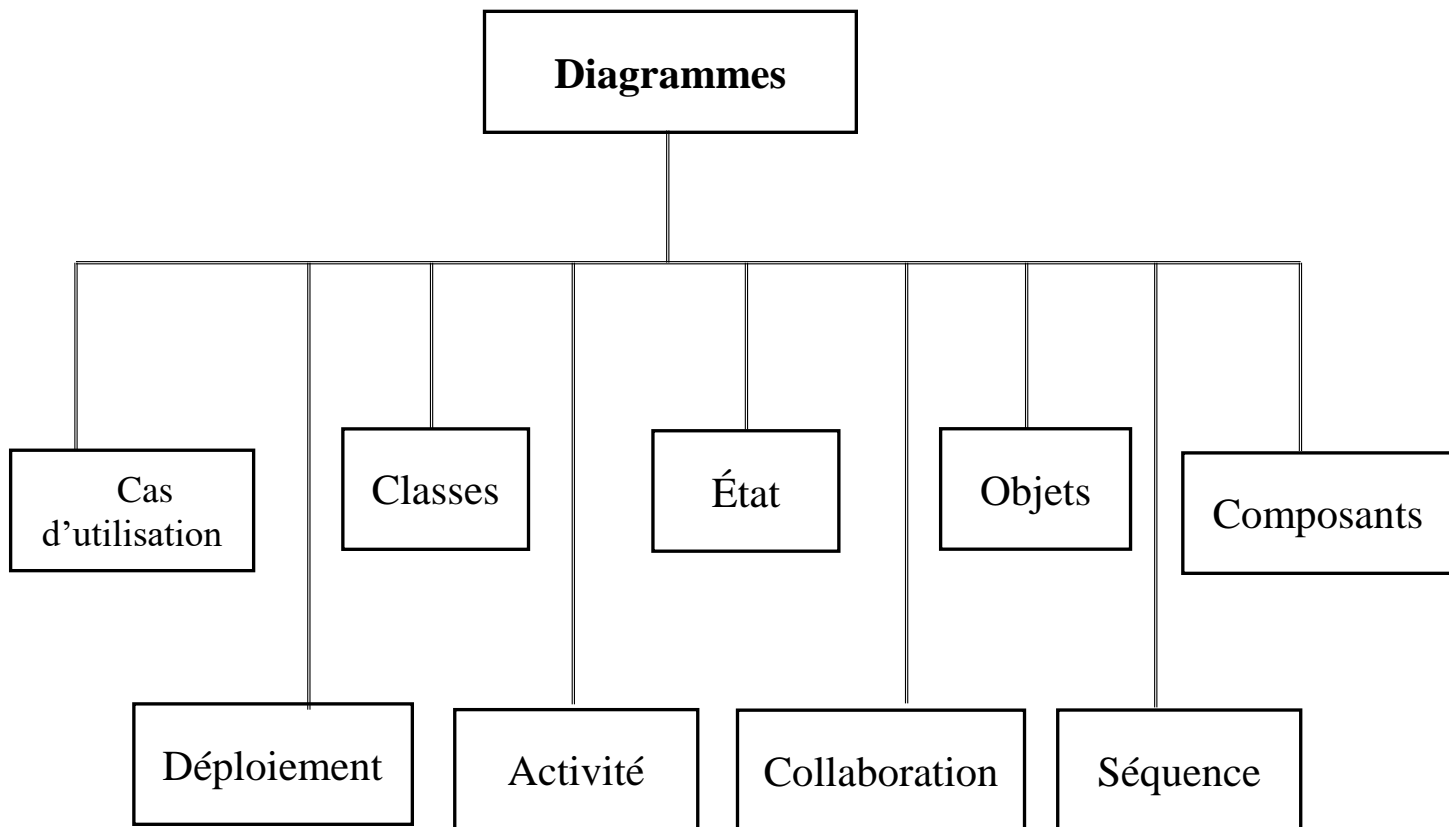


FIGURE 2 - les différents diagrammes UML

### 2.2.2. La technologie orienté objet

La programmation orientée objet (POO), ou programmation par objet, est un paradigme de programmation informatique, Il consiste en la définition et l'interaction de briques

logicielles appelées objets. Un objet représente un concept, une idée ou toute entité du monde physique, comme une voiture, une personne ou encore une page d'un livre. Il possède une structure interne et un comportement, et il sait interagir avec ses pairs. Il s'agit donc de représenter ces objets et leurs relations ;

L'interaction entre les objets via leurs relations permet de concevoir et réaliser les fonctionnalités attendues, de mieux résoudre le ou les problèmes. Dès lors, l'étape de modélisation revêt une importance majeure et nécessaire pour la POO. C'est elle qui permet de transcrire les éléments du réel sous forme virtuelle.

Orthogonalement à la programmation par objet, afin de faciliter le processus d'élaboration d'un programme, existent des méthodologies de développement logiciel objet dont la plus connue est le Unified Software Development Process (UP) qui utilisent des langages de modélisation tels que le Unified Modeling Language (UML). Même s'il est possible de concevoir par objets une application informatique sans utiliser des outils logiciels dédiés, ces derniers facilitent la conception, la maintenance, et la productivité.

On en distingue plusieurs sortes :

- Les langages de programmation (Java, Python PHP...)
- les outils de modélisation qui permettent de concevoir sous forme de schémas semi-formels la structure d'un programme (UMLDraw, DBDesigner...)
- les ateliers de génie logiciel ou AGL (Visual Studio pour des langages Dotnet, NetBeans ou Eclipse pour le langage Java).

Bref la technologie objet est la voie la plus prometteuse pour déployer des systèmes basés sur des composants qui peuvent être adaptés et changés sans avoir à examiner la totalité du code.

### 2.2.3. L'architecture MVT (Modèle – Vue – Template)

Le MVT représente une architecture orientée autour de trois pôles : le modèle, la vue et le template. Elle s'inspire de l'architecture MVC (Modèle Vue Contrôleur), très répandue dans les Framework web. Son objectif est de séparer les responsabilités de chaque pôle afin que chacun se concentre sur ses tâches. Elle offre aussi un cadre pour structurer une application. Dans l'architecture MVT, les rôles des trois entités sont les suivantes :

- Modèle
- Vue
- Template

### **A. Rôle du modèle**

Le modèle contient les données manipulées par le programme. Il assure la gestion de ces données et garantit leur intégrité. Dans le cas typique d'une base de données, c'est le modèle qui la contient.

Le modèle offre des méthodes pour mettre à jour ces données (insertion suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ses données. Dans le cas de données importantes, le modèle peut autoriser plusieurs vues partielles des données. Si par exemple le programme manipule une base de données pour la gestion des effectifs, le modèle peut avoir des méthodes pour avoir l'effectif total, toutes les notes des élèves ainsi que leur différent bulletin.

### **B. Rôle de la vue**

La vue fait l'interface avec l'utilisateur. Sa première tâche est d'afficher les données qu'elle a récupérées auprès du modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d'une entrée, boutons, ... ). Ses différents événements sont envoyés au template.

La vue peut aussi donner plusieurs vues, partielles ou non, des mêmes données. Par exemple, l'application de conversion de bases a un entier comme unique donnée. Ce même entier est affiché de multiples façons (en texte dans différentes bases, bit par bit avec des boutons à cocher, avec des curseurs). La vue peut aussi offrir la possibilité à l'utilisateur de changer de vue.

### **C. Rôle du template**

Un template se compose de parties statiques de la sortie HTML souhaitée ainsi que d'une syntaxe spéciale décrivant comment le contenu dynamique sera inséré. Le template est chargé de la synchronisation du modèle et de la vue. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le template demande la modification des données au modèle et ensuite avertit la vue que les données ont changé pour que celle-ci se mette à jour. Certains événements de l'utilisateur ne

concernent pas les données mais la vue. Dans ce cas, le template demande à la vue de se modifier.

#### D. Les interactions

Les différentes interactions entre le modèle, la vue et le template sont résumées par le schéma de la figure suivante .

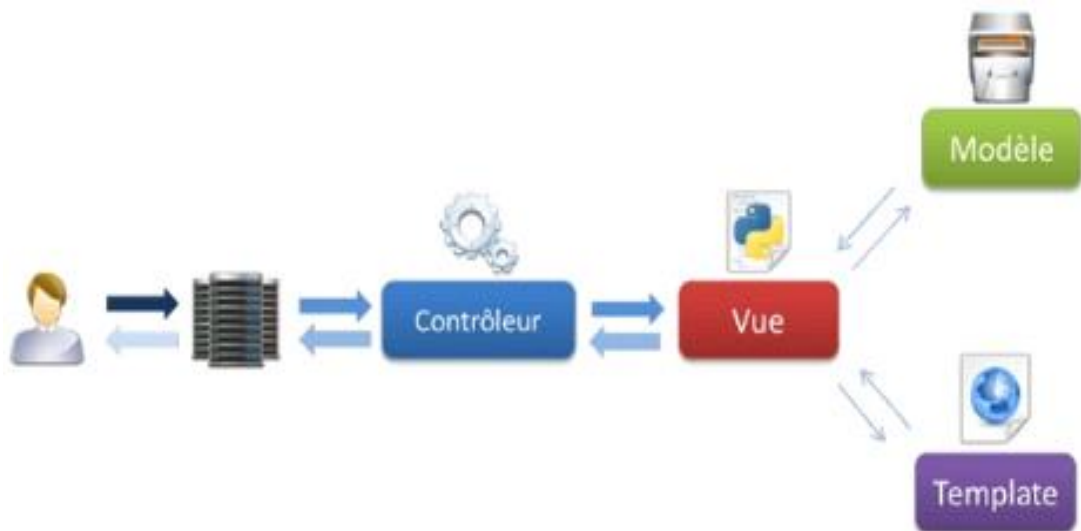


FIGURE 3 - Interactions entre le modèle, la vue et le template.

#### 2.2.4. Le Framework Django

En programmation informatique, un framework (appelé aussi cadre applicatif, cadre d'applications ou encore infrastructure de développement ) désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture).

Les frameworks sont conçus et utilisés pour modeler l'architecture des logiciels applicatifs, des applications web, des middlewares et des composants logiciels. Les frameworks sont acquis par les informaticiens, puis incorporés dans des logiciels applicatifs mis sur le marché, ils sont par conséquent rarement achetés et installés séparément par un utilisateur final.

### 2.3. Les langages utilisés

#### 2.3.1. HTML



L'Hypertext Markup Language, abrégé HTML ,est un langage conçu pour représenter les pages web. Ce langage de balisage permet de structurer sémantiquement et de mettre en forme le contenu des pages ,d'inclure des ressources multimédia dont des images des formulaires, de saisie ...

### 2.3.2. CSS

De l'anglais Cascading Style Sheets (les feuilles de style en cascade),le CSS constitue un langage informatique permettant d'ajouter des contraintes de mise en forme graphique dans des scripts dans différents langages en les interconnectant.

### 2.3.3. Python

Python est un langage de programmation, comme C, PHP, etc... Il est interpréter et non compilé l'utilisateur ne doit pas compiler son code Python avant de l'exécuter. Python est par ailleurs un langage qui permet l' utilisation interactive. C'est aussi est un langage orienté objet, au typage dynamique (le type d'un objet peut changer au cours d'un programme).

### 2.3.4. JS

JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page web. À chaque fois qu'une page web fait plus que simplement afficher du contenu statique — afficher du contenu mis à jour à des temps déterminés, des cartes interactives, des animations 2D/3D, des menus vidéo défilants, ou autre, JavaScript a de bonnes chances d'être impliqué. C'est la troisième couche des technologies standards du web, les deux premières (HTML et CSS) étant couvertes bien plus en détail dans d'autres tutoriels sur MDN.

L'étude technique et fonctionnelle nous a permis de passer en revue les besoins fonctionnels et non fonctionnels de notre projet ainsi que les méthodes et technique choisies pour l'élaboration de notre application. Maintenant, nous allons passer à l'analyse et la conception de notre projet avec le langage UML.

### 3. ANALYSE ET CONCEPTION

nous allons maintenant exprimer les besoins sous forme de diagrammes de cas d'utilisations.

#### 3.1. Modélisation en langage UML

UML (Unified modeling Language) se définit comme des langages de modélisation graphique et textuelle destinés à comprendre et décrire des besoins, spécifier, concevoir des solutions et communiquer des points de vue. Il unifie à la fois les notations et les concepts orientés objet. UML propose un ensemble de diagrammes, en voici quelques-uns :

##### 3.1.1. Identification des acteurs

Les acteurs sont des entités externes qui interagissent avec le système, comme une personne humaine ou un robot. Une même personne (ou robot, ...) peut-être plusieurs acteurs pour un système, c'est pourquoi les acteurs doivent surtout être décrits par leur rôle, ce rôle décrit les besoins et les capacités de l'acteur. Un acteur agit sur le système (accès de lecture/écriture). L'activité du système a pour objectif de satisfaire les besoins de l'acteur. Les acteurs sont représentés par un pictogramme humanoïde sous-titré par le nom de l'acteur. Dans notre application, nous avons défini trois acteurs qui sont :

- Elève  
Rôle : consulter son relevé de note
- Enseignant  
Rôle : gérer les moyennes des élèves, consulter sa fiche de matières
- Administrateur  
Rôle : valider les moyennes, générer les bulletins des élèves, gérer les enseignants et les élèves, gérer les données de base (école, filières, niveau, classe, élèves, enseignants, matières...)
- Administrateur principal Rôle : gérer les utilisateurs

##### 3.1.2. Le diagramme de cas d'utilisation

Le diagramme de cas d'utilisation a pour but de donner une vision globale sur les interfaces de notre future application. C'est le premier diagramme UML constitué d'un

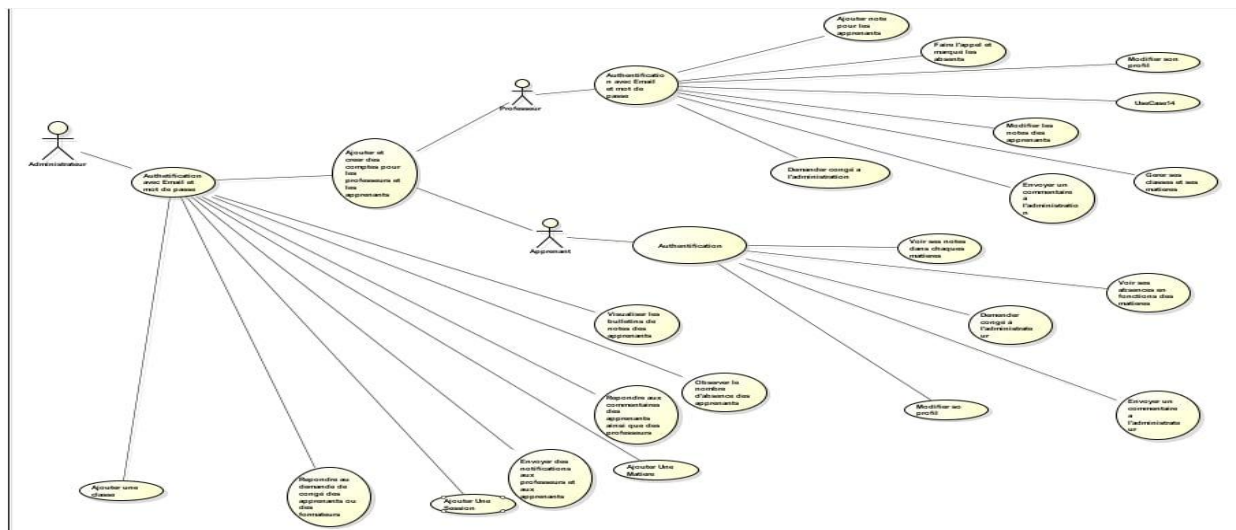
ensemble d'acteurs qui agit sur des cas d'utilisations et qui décrit sous la forme d'actions et de réactions, le comportement d'un système du point de vue utilisateur.

Un acteur est un utilisateur qui communique et interagit avec les cas d'utilisation du système. C'est une entité ayant un comportement comme une personne, système ou une entreprise.

Le système fixe les limites en relation avec les acteurs qui l'utilisent (en dehors du système) et les fonctions qu'il doit fournir (à l'intérieur du système).

Un cas d'utilisation représente un ensemble de séquences d'actions à réaliser par le système et produisant un résultat observable intéressant pour un acteur particulier représenté par des ellipses et limité par un rectangle pour représenter le système.

Chaque usage que les acteurs font du système est représenté par un cas d'utilisation. Chaque cas d'utilisation représente une fonctionnalité qui leur est offerte afin de produire le résultat attendu. Ci-après le diagramme de cas d'utilisation général de notre projet :



**FIGURE 4 - Diagramme de cas d'utilisation générale.**

### Description des cas d'utilisations

Les différents cas d'utilisations de notre application ont été décrits dans les tableaux ci-après en fonction de l'intitulé du cas.

- S'authentifier

**Objectif :** Permet l'identification de chaque utilisateur lui donnant accès aux fonctionnalités propices.

**Acteurs principaux :** l'administrateur, les enseignants

**Pré condition :** L'utilisateur dispose d'un compte d'accès au système.

**Description :** L'utilisateur saisi son identifiant et son mot de passe. Il est reconnu par le système. Il peut accéder en fonction de son profil utilisateur.

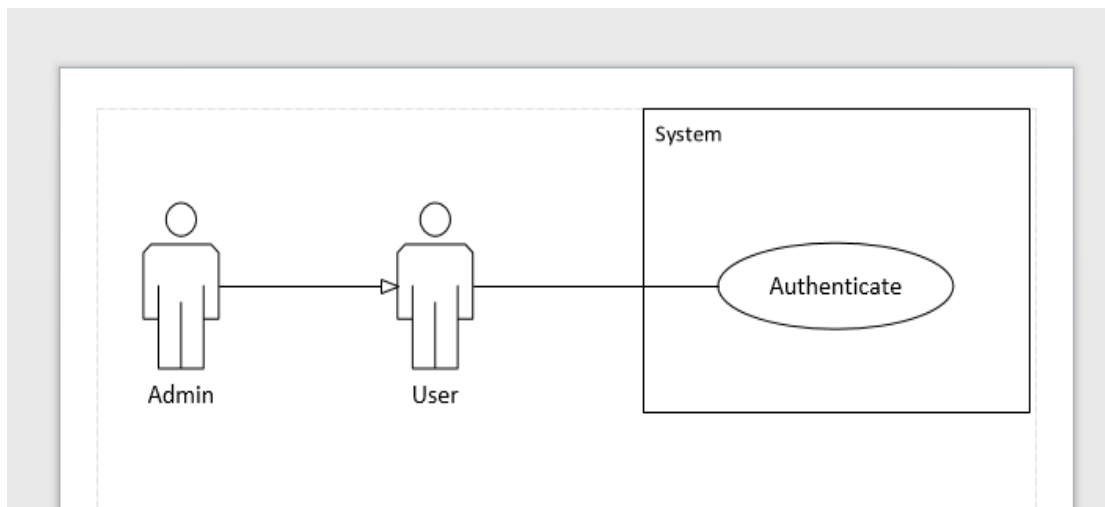


FIGURE 5 - Diagramme de cas d'utilisation «s'authentifier »

- Paramétrage

**Objectif :** Ajouter, modifier et supprimer certaines informations concernant les classes, les matières, les niveaux et les filières. Permettre la réconduite de toutes les données à la création d'une nouvelle année universitaire. Permettre également la gestion des utilisateurs de l'application.

**Acteurs principaux :** l'administrateur

**Pré condition :** l'utilisateur doit être connecté au système

**Description :** A la création d'une nouvelle année académique, les matières, les classes et les niveaux de l'année clôturée sont recopier automatiquement pour la nouvelle année. Une vérification manuelle est faite par le l'administrateur pour reconduire ou supprimer certaines données qui ne sont pas autorisées pour la nouvelle année.

L'administrateur peut créer un nouveau compte utilisateur, consulter, modifier et supprimer un utilisateur. En plus d'une liste de tous les classes autorisés pendant une année académique, il a également la possibilité de les modifier et les supprimer.

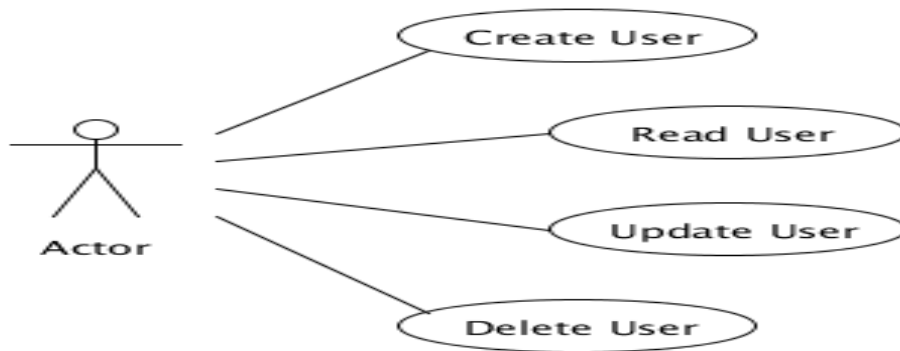


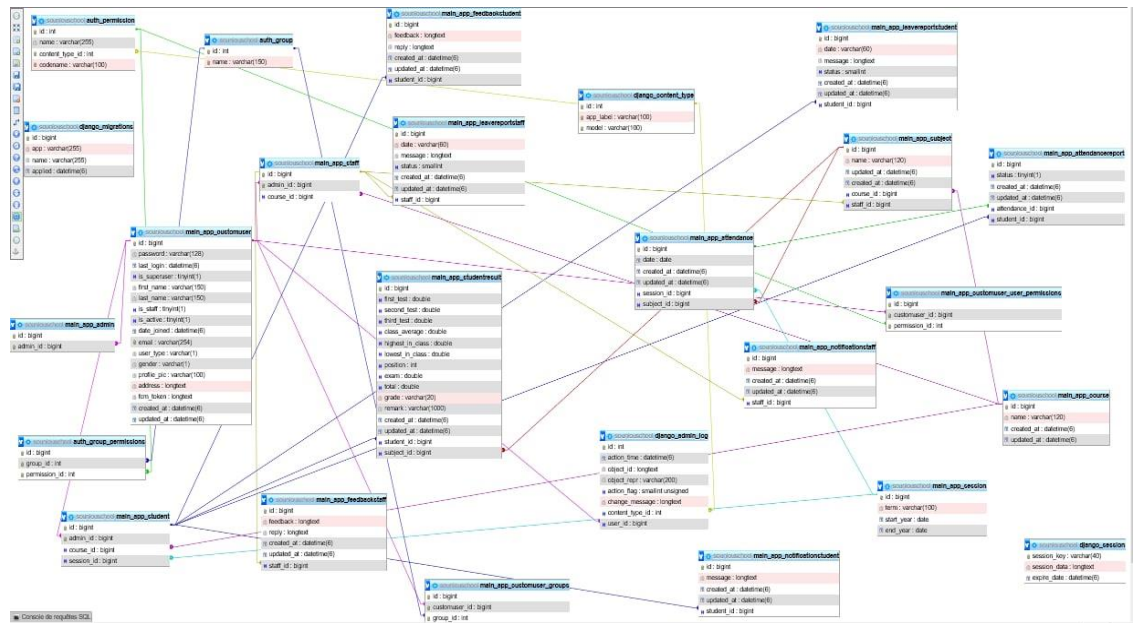
FIGURE 6 - Diagramme de cas d'utilisation «paramétrage »

## 3.2.Mise en place de la base donnés

### 3.2.1. Modèle Physique de données

Dans la méthode Merise, le modèle physique des données (MPD) consiste à implanter une base de données dans un SGBDR (Système de Gestion de Base de Données Relationnelles). Le langage utilisé pour ce type d'opération est le SQL. On peut également faire usage d'un AGL (PowerAMC, \i\TinDesign, etc.) qui permet de générer automatiquement la base de données.

Un modèle Physique de Données est une étape de définition des données à l'intérieur de la structure physique de l'ordinateur c'est-à-dire le résultat de la décision technique qui a été prise en fonction des objets et des contraintes techniques. C'est un formalisme qui permet de préciser le système de stockage employé pour un système de gestion de base de données. Ci-après le MPD du projet :



### FIGURE 7 - Modèles Physiques des Données

Dans ce chapitre , nous avons présenté les différents diagrammes de cas d'utilisations définis qui ont permis de bien comprendre les besoins du système à développer ainsi que les différentes interactions entre les objets participant à son fonctionnement, chose qui facilitera la phase d'implémentation.

## 4. IMPLÉMENTATION ET RÉSULTAT

### 4.1. Plateforme de développement

#### 4.1.1. SGBDR MySQL

- **Définition**

**MySQL** est une **base de données** relationnelle libre qui a vu le jour en 1995 et très employée sur le Web, souvent en association avec *PHP* (langage) et *Apache* (serveur web). MySQL fonctionne indifféremment sur tous les systèmes d'exploitation (*Windows, Linux, Mac OS* notamment).

Il est basé sur le langage de requête de structure ([SQL](#)), utilisé pour ajouter, supprimer et modifier des informations dans la base de données.

- **Avantages**

MySQL est beaucoup plus simple à utiliser que la plupart des serveurs de bases de données commerciaux.

Il dispose d'un système de sécurité permettant de gérer les personnes et les machines pouvant accéder aux différentes bases.

MySQL tourne sur divers systèmes tels que Unix, Windows, Linux ou OS/2.

Les sources étant fournies, il est possible d'améliorer MySQL.

- **Inconvénients**

Dans une base de données relationnelles, les identifiants (clefs primaires) d'une table se retrouvent en clefs étrangères d'une autre table en relation avec la première.

Ce système de clefs primaires et clefs secondaires permet d'assurer l'intégrité des données.

En effet, lorsqu'on modifie une clef primaire, les clefs secondaires sont automatiquement mises à jour.

MySQL ne supporte pas ce système.

Lorsqu'on crée une table, on peut très bien utiliser l'argument FOREIGN KEY (clef étrangère) pour un attribut mais ceci n'a aucune influence.

Cet argument est juste là pour assurer une meilleure importation de bases gérées par un autre SGBDR que MySQL.

#### 4.1.2. Le langage de développement Python

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages.

Il reste aussi accessible pour les débutants, à condition de lui consacrer un peu de temps pour la prise en main. De nombreux tutoriels sont d'ailleurs disponibles pour l'étudier sur des sites Internet spécialisés ou sur des comptes YouTube. Sur les forums d'informatique, il est toujours possible de trouver des réponses à ses questions, puisque beaucoup de professionnels l'utilisent.

A quoi sert le langage Python ?

Les principales utilisations de Python par les développeurs sont :

- la programmation d'applications
- la création de services web
- la génération de code
- la méta programmation.

Techniquement, ce langage servira surtout pour le scripting et l'automatisation (interaction avec les navigateurs web).

#### 4.2. Les interfaces de l'application

L'application se compose de plusieurs interfaces qui guident l'administrateur, l'enseignant et l'élève vers les différentes fonctions de l'application après son authentification.

- Authentification

Cet interface permet à l'utilisateur de s'authentifier et de se connecter au tableau de bord de l'application. L'utilisateur doit entrer son identifiant et son mot de passe pour accéder à



l'application. En cas d'échec un message d'alerte s'affiche. En cas de succès, l'utilisateur accède à la page d'accueil de l'application. Cet interface est la page d'accueil de notre application, Il propose tous les modules et les liens de navigation rapides à l'utilisateur connecté .

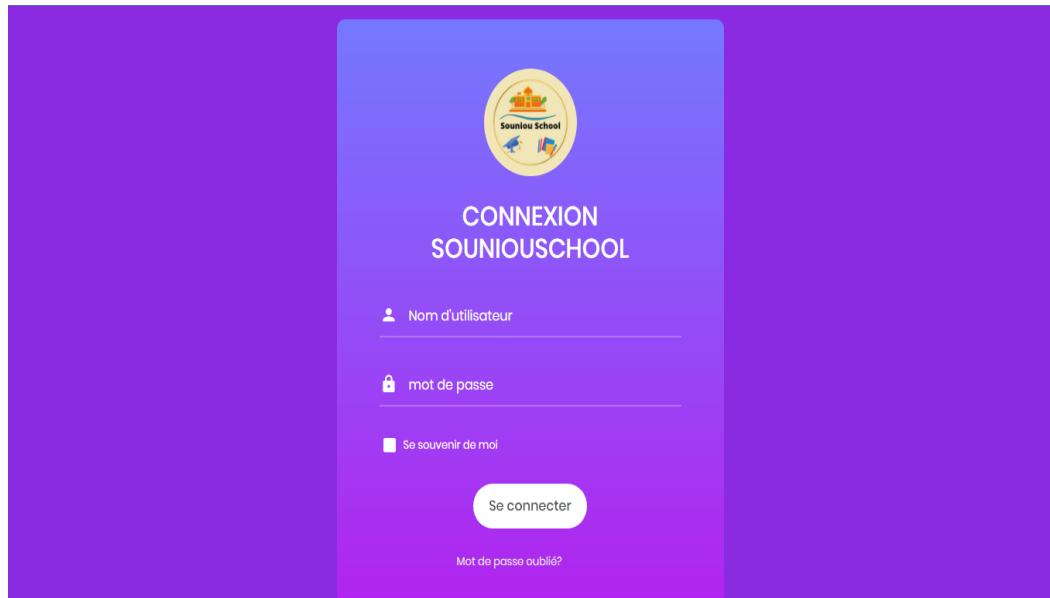
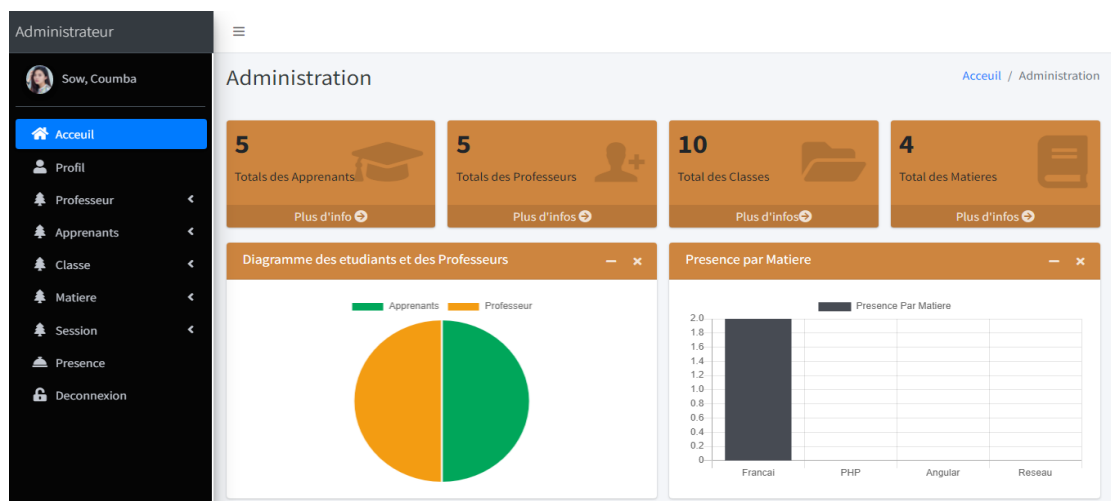


FIGURE 8 – Interface 1 « d'authentification »

#### 4.2.1. Administration

Cette section consiste à gérer les utilisateurs de notre application, dans notre cas il s'agit des enseignants. Chaque enseignant devra disposer d'un login et un mot de passe ce qui lui permettra d'accéder à sa partition. cette étape est effectuée par l'administrateur.



**FIGURE 9 – Interface 2 « Administrateur »**

Ensuite nous avons la partie profil qui permet à l'utilisateur de modifier ces Identifiants et l'image de son profil.

The screenshot shows the 'Vue/Editer Profil' (View/Edit Profile) page for an administrator. The left sidebar contains a menu with options: Accueil, Profil (selected), Professeur, Apprenants, Classe, Matiere, Session, Presence, and Deconnexion. The main content area displays the profile of 'Coumba Sow', an administrator of the establishment. It includes a profile picture, name, email (omarthioms35@gmail.com), gender (F), and a phone number (77 872 40 12). To the right, there is a form to edit the profile with fields for First name, Last name, Email, Gender, Password, and Profile pic.

**FIGURE 10 – Interface « Éditer profil »**

Cette page permet d'ajouter un nouveau professeur en même temps lui créer un compte dans l'application.

The screenshot shows the 'Ajouter Professeur' (Add Teacher) page. The left sidebar is the same as in Figure 10, but the 'Professeur' option is selected. The main content area displays a form to add a new teacher. It includes fields for First name, Last name, Email, Gender, Password, and Profile pic. The email field is pre-filled with 'omarthioms35@gmail.com' and has a red error message below it: 'Cette Adresse Email existe deja' (This email address already exists). The gender dropdown is set to 'Masculin'.

**FIGURE 11 – Interface « Ajouter professeur »**

Dans cette partie nous avons la liste des professeurs de l'établissement en ayant la possibilité de modifier ou de supprimer

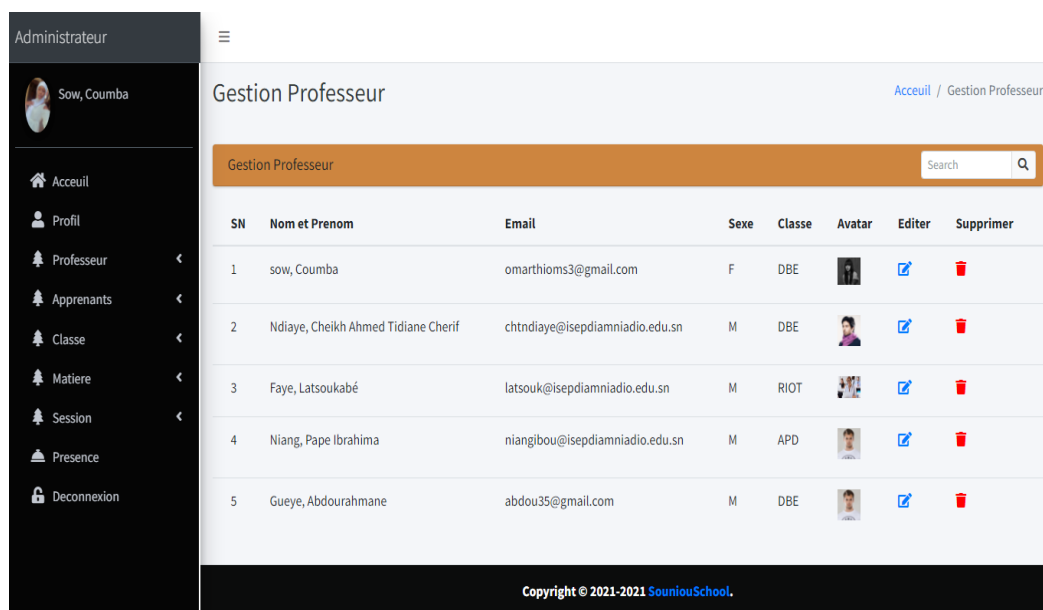


FIGURE 12 – Interface « Gestion professeur »

Cette permet à l'administrateur d'envoyer des notifications aux professeurs en cas de besoin.

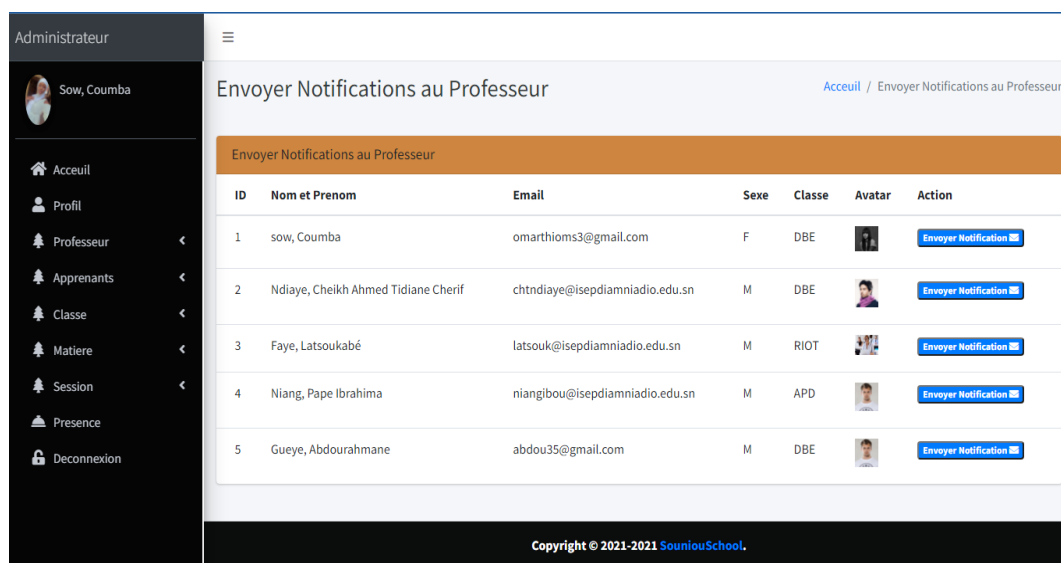
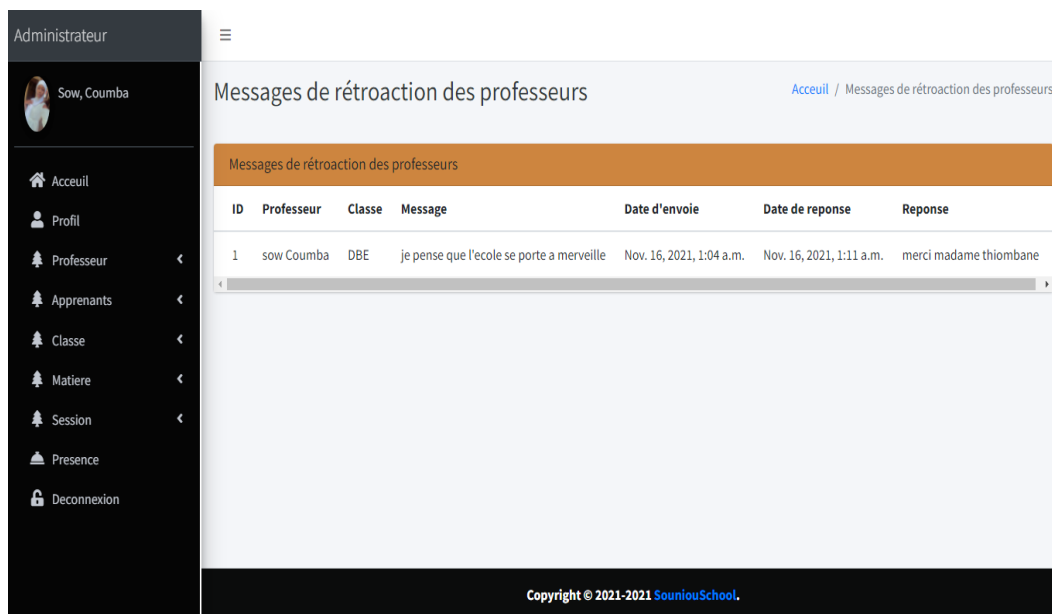


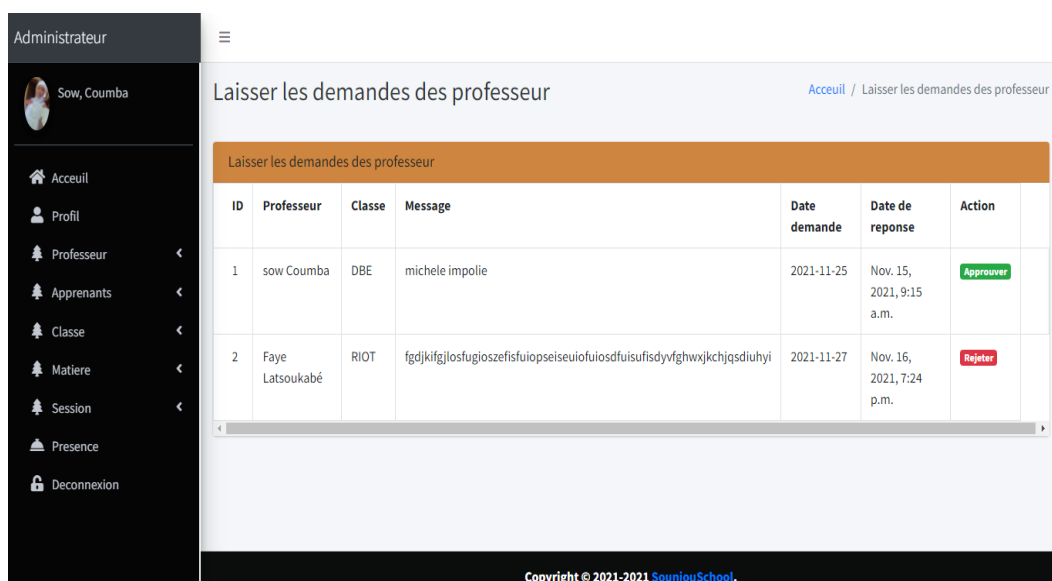
FIGURE 13 – Interface « Envoyer notification au professeur »

Ici nous avons le commentaire des professeurs et l'administrateur a le droit de les répondre.



**FIGURE 14 – Interface « Message de rétroaction des professeurs »**

Dans cette page nous avons tous les congés demander par les professeurs et s'ils sont accordés où refuser.



**FIGURE 15 – Interface « Demande des professeurs »**

Dans cette partie l'administrateur peut ajouter un nouvel apprenant en même temps lui créer un compte pour qu'il se connecte dans l'application.

FIGURE 16 – Interface « Ajouter Apprenants »

Ici nous avons la liste des apprenants avec leur filière avec la possibilité de modifier ou de supprimer.

ID	Nom et Prenom	Email	Sexe	Classe	Avatar	Editer	Supprimer
1	Thiam, Mamadou	meumzo35@gmail.com	M	DBE			
2	Kane, Ndiogou	ndioks35@gmail.com	M	DBE			
3	Gaye, Mouhamed	mouha35@gmail.com	M	DBE			
4	Diallo, Souleymane	soul35@gmail.com	M	DBE			
5	Fam, Magueye	lamp35@gmail.com	M	DBE			

Copyright © 2021-2021 SouniouSchool.

. FIGURE 17 – Interface « Gestion Apprenants »

Ainsi dans cette partie nous avons le bulletin de note de l'apprenants avec le rang, la moyenne, l'appréciation etc.

**Administrateur**  
Sow, Coumba

**Voir Resultat** [Accueil](#) / [Voir Resultat](#)

**"SOUNIOU SCHOOL"**  
IA DES ALMADIES - COMMUNE DE OUAKAM  
Tel: 33 872 40 12  
Respect, Reussite, Discipline

Totals: 92.0

Nom: Fam Prenom: Magueye  
Classe: DBE Moyenne:  
Debut: Nov, 16, 2021 Fin: June 16, 2022

ID	Sujet	Note 1	Note 2	Note 3	Examen	Total	Moyenne Classe	Moyenne du Premier	Moyenne Du Dernier	Rang	Grade	Appreciation
1	Francais	16.0	15.0	17.0	15.0	92.0	15.0	19.0	6.0	2	Tres bien	eleve serieux a encourager

. FIGURE 18 – Interface « Voir Résultat »

Dans cette partie l'administrateur peut envoyer des notifications aux apprenants.

**Administrateur**  
Sow, Coumba

**Envoyer Notifications a l'eleve** [Accueil](#) / [Envoyer Notifications a l'eleve](#)

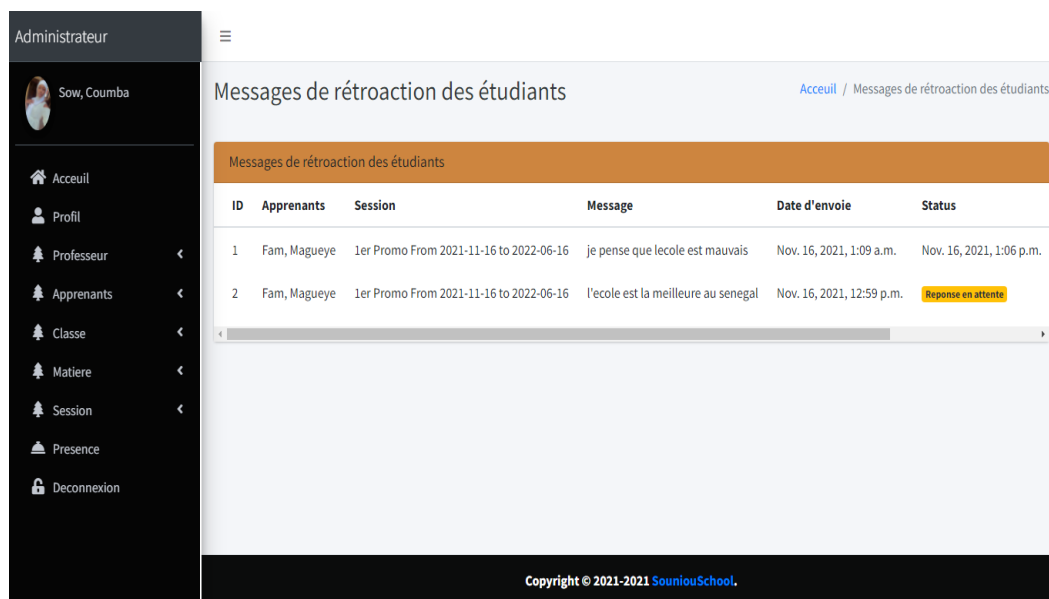
Envoyer Notifications a l'eleve

ID	Nom et Prenom	Email	Sexe	Classe	Avatar	Action
1	Thiam, Mamadou	meumzo35@gmail.com	M	DBE		<a href="#">Envoyer Notification</a>
2	Kane, Ndiogou	ndioks35@gmail.com	M	DBE		<a href="#">Envoyer Notification</a>
3	Gaye, Mouhamed	mouha35@gmail.com	M	DBE		<a href="#">Envoyer Notification</a>
4	Diallo, Souleymane	soul35@gmail.com	M	DBE		<a href="#">Envoyer Notification</a>
5	Fam, Magueye	lamp35@gmail.com	M	DBE		<a href="#">Envoyer Notification</a>

Copyright © 2021-2021 SouniouSchool.

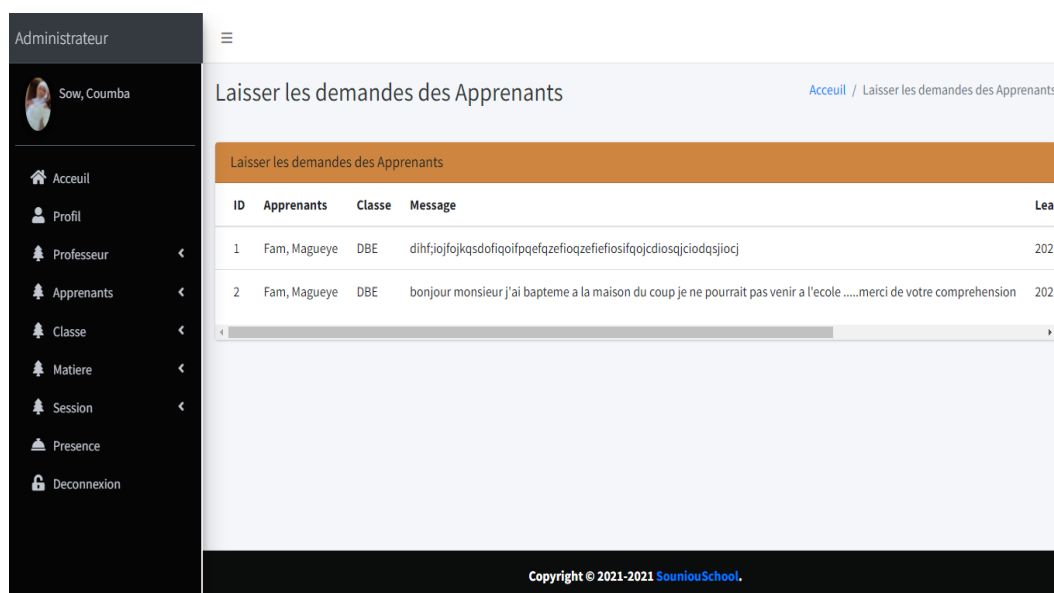
FIGURE 19 – Interface « Envoyer notification à l'élève »

Cette partie permet à l'administrateur de voir le commentaire des élèves et de les répondre.



**FIGURE 20 – Interface « Message de rétroaction des élèves »**

L'administrateur reçoit les demandes de congés des apprenants en ayant la possibilité d'accepter ou de refuser



**FIGURE 21 – Interface « Lister demande élèves »**

Cette partie permet à l'administrateur d'ajouter de nouvelles classes.

FIGURE 22 – Interface « Ajouter classes »

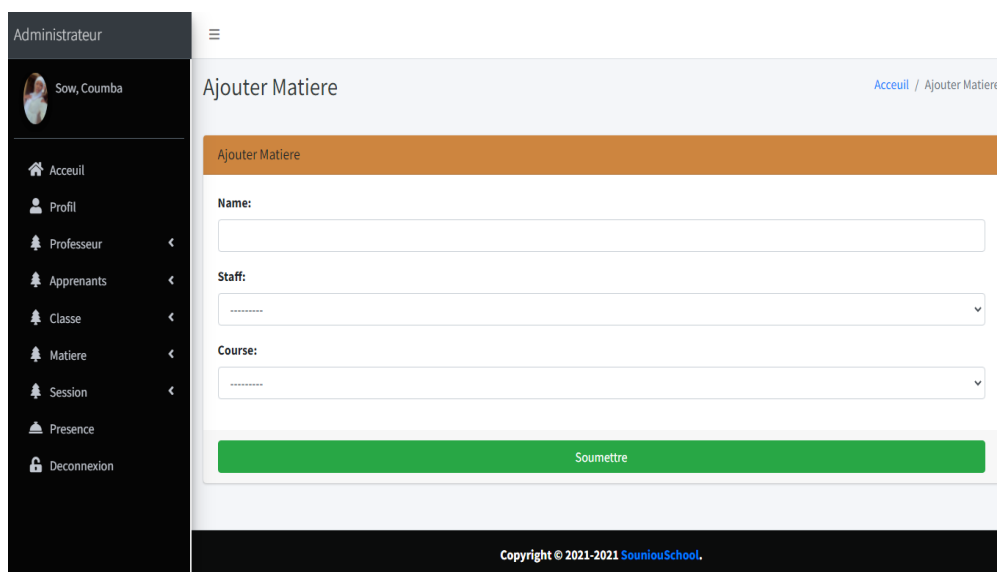
Ici nous avons les différents classes ou filières dans l'établissement et l'administrateur peut supprimer ou modifier

ID	Nom de la classe	Editer	Supprimer
1	DBE		
2	DFE		
3	APD		
4	ABD		
5	RIOT		
6	GDE		
7	EEA		
8	CPA		

FIGURE 23 – Interface « Gestion classe »



Dans cette partie l'administrateur peut ajouter une matière en choisissant le professeur et le classe.



Administrateur

Sow, Coumba

Accueil / Ajouter Matière

Ajouter Matière

Name:

Staff:

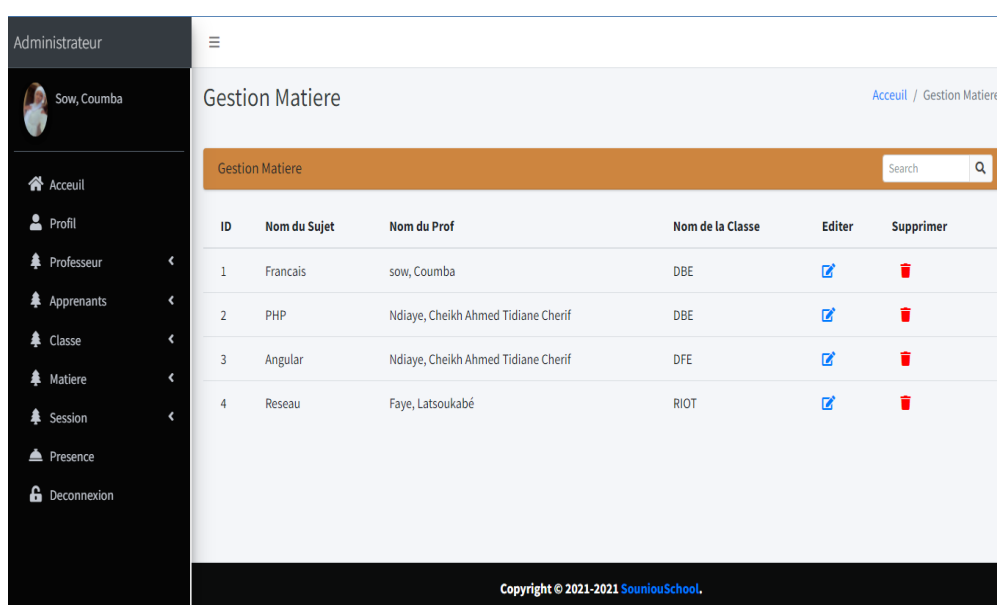
Course:

Soumettre

Copyright © 2021-2021 SouniouSchool.

FIGURE 24 – Interface « Ajouter Matière »

Dans partie nous avons les matières et les professeurs qui le gère ainsi la classe que le cours se passe.



Administrateur

Sow, Coumba

Accueil / Gestion Matière

Gestion Matière

ID	Nom du Sujet	Nom du Prof	Nom de la Classe	Editer	Supprimer
1	Francais	sow, Coumba	DBE		
2	PHP	Ndiaye, Cheikh Ahmed Tidiane Cherif	DBE		
3	Angular	Ndiaye, Cheikh Ahmed Tidiane Cherif	DFE		
4	Reseau	Faye, Latsoukabé	RIOT		

Copyright © 2021-2021 SouniouSchool.

FIGURE 25 – Interface « Gestion Matière »

Dans cette partie l'administrateur peut ajouter une session avec un nom, une date début et une date de fin.

Administrateur

Sow, Coumba

Accueil

Profil

Professeur

Apprenants

Classe

Matiere

Session

Presence

Deconnexion

Ajouter Session

Term:

Start year:

jj/mm/aaaa

End year:

jj/mm/aaaa

Soumettre

Copyright © 2021-2021 SouniouSchool.

FIGURE 26 – Interface « Ajouter Session »

Cette page liste les différentes sessions existantes avec la possibilité de modifier ou de supprimer

Administrateur

Sow, Coumba

Accueil

Profil

Professeur

Apprenants

Classe

Matiere

Session

Presence

Deconnexion

Gestion Session

Search

ID	Teme	Debut	Fin	Editer	Supprimer
1	1er Promo	Nov. 16, 2021	June 16, 2022		

Copyright © 2021-2021 SouniouSchool.

FIGURE 27 – Interface « Gestion Session »

Ici nous avons les absences et présences des apprenants en fonction de la session, de la matière et de la date de déroulements des cours.

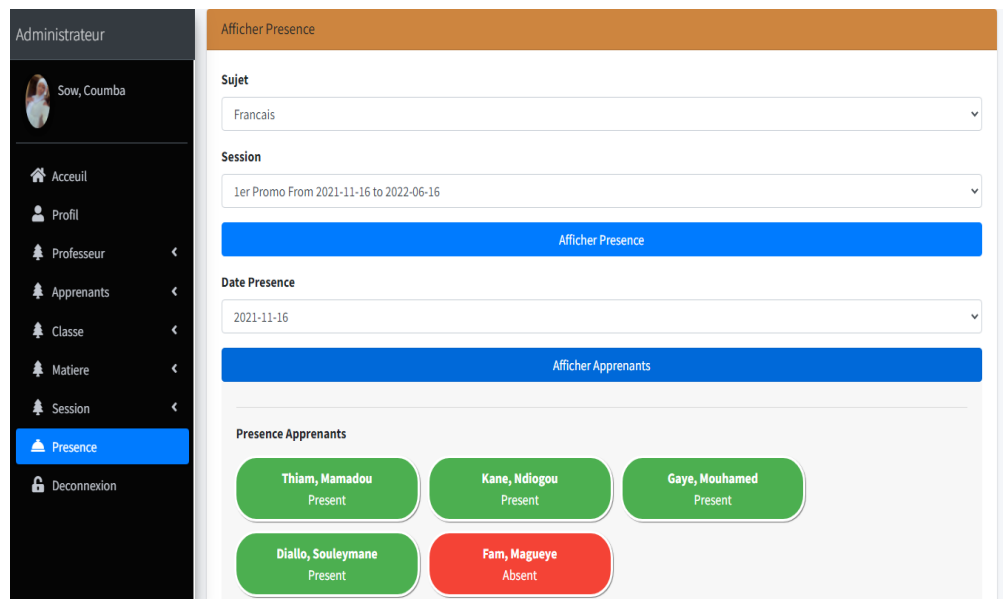


FIGURE 28 – Interface « Afficher Présence »

Maintenant nous sommes dans l'accueil de la partie des professeurs, Avec le nombre d'apprenants qu'il gère le total des absences le nombre de congé accorder et les matières qu'il dirige dans l'établissement, ainsi le diagramme des absences, des congés et absences.

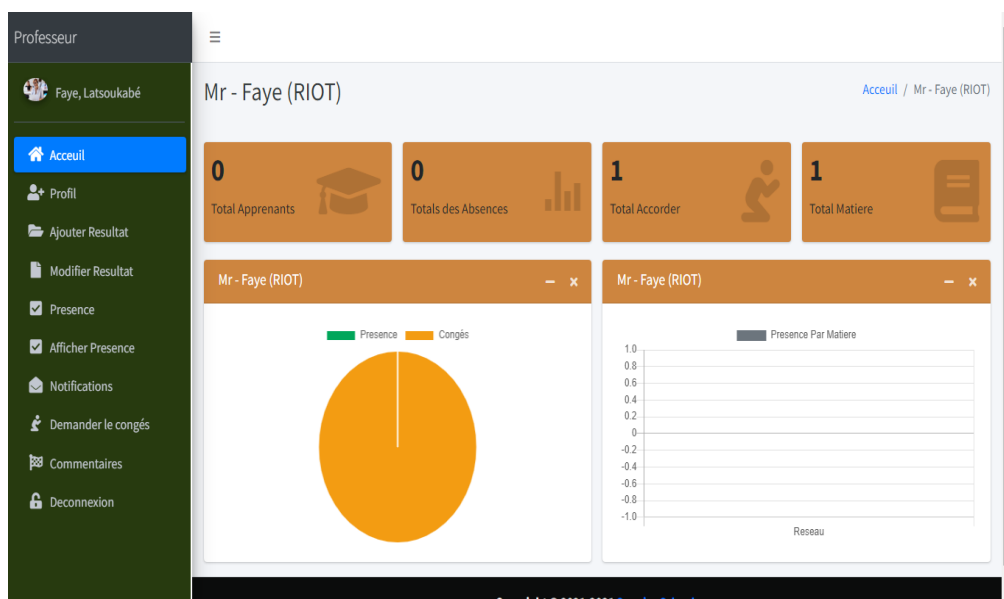


FIGURE 29 – Interface « Accueil professeur »

Ici le professeur peut modifier son profil c'est-à-dire ses identifiants et son image profil.

The screenshot shows the 'Vue/Mise à jour Profil' (View/Update Profile) page for a teacher named Faye, Latsoukabé. The sidebar on the left contains navigation links: Accueil, Profil (highlighted), Ajouter Resultat, Modifier Resultat, Presence, Afficher Presence, Notifications, Demander le congés, Commentaires, and Deconnexion. The main content area has a header 'Vue/Mise à jour Profil' and a sub-header 'Vue/Mise à jour Profil'. Below the header, there is a profile card showing the teacher's name, email, and a button to update the profile. The main form contains fields for First name, Last name, Email, Gender, Password, and Profile pic.

FIGURE 30 – Interface « Vue/Mise à jour » professeur

Cette partie permet aux professeurs d'ajouter les notes des apprenants en fonction de la session et de la matière.

The screenshot shows the 'Téléchargement des résultats' (Download Results) page for a teacher named Faye, Latsoukabé. The sidebar on the left contains navigation links: Accueil, Profil, Ajouter Resultat (highlighted), Modifier Resultat, Presence, Afficher Presence, Notifications, Demander le congés, Commentaires, and Deconnexion. The main content area has a header 'Téléchargement des résultats' and a sub-header 'Téléchargement des résultats'. Below the header, there is a form for selecting the subject and session, and a table for displaying the results of the students.

FIGURE 31 – Interface « Téléchargement des résultats »

Dans cette partie le professeur peut modifier les notes des apprenants en fonction de la session et de la matière.

The screenshot shows the 'Modifier Resultat' (Edit Result) interface. On the left is a dark green sidebar with a menu: Accueil, Profil, Ajouter Resultat, Modifier Resultat (highlighted), Presence, Afficher Presence, Notifications, Demander le congés, Commentaires, and Deconnexion. The main area is titled 'Editer Resultat' and contains a form with the following fields: Session Year (dropdown), Subject (dropdown), Student (dropdown), First test (text input), Second test (text input), Third test (text input), Exam (text input), Total (text input), Class average (text input), Highest in class (text input), Lowest in class (text input), Position (text input), Grade (dropdown), and Remark (text area). A blue button at the bottom right says 'Mettre à jour Resultat'.

FIGURE 32 – Interface « Éditer résultats »

Dans cette partie le professeur peut faire l'appel et marquer les élèves Absent ou présent.

The screenshot shows the 'Demande' (Request) interface. The sidebar is the same as in Figure 32, with 'Presence' highlighted. The main area is titled 'Demande' and contains a form with the following fields: Sujet (dropdown), Session de l'année (dropdown), a large orange button labeled 'Faire l'appel', Date Presence (text input with a calendar icon), and a list of students with checkboxes: sy habib (checked), tall assane (unchecked), and sow marie (checked). A green button at the bottom left says 'Sauvegarder Presence'.

FIGURE 33 – Interface « Demande d'autorisation d'absence »

Cette permet au professeur de rectifier l'absence ou la présence d'un apprenant en fonction de la session et de la date et du cours.

The screenshot shows the 'Mettre à jour' (Update) interface for a teacher. The sidebar on the left includes options like 'Accueil', 'Profil', 'Ajouter Resultat', 'Modifier Resultat', 'Presence', 'Mettre a Presence' (highlighted), 'Notifications', 'Demander le congés', 'Commentaires', and 'Deconnexion'. The main form has the following sections:

- Mettre à jour** (orange header)
- Sujet**: A dropdown menu with 'Reseau' selected.
- Session**: A dropdown menu with '1er Promo From 2021-11-16 to 2022-06-16' selected.
- Afficher Presence** (orange button)
- Date Presence**: A dropdown menu with '2021-11-29' selected.
- Afficher Apprenants** (orange button)
- Apprenants Presence**: A section with three checkboxes: 'sy habib [Absent]' (unchecked), 'tall assane [Present]' (checked), and 'sow marie [Present]' (checked).
- Sauvegarder Presence** (green button)

FIGURE 34 – Interface « Mise à jour absence »

Cette partie permet au professeur de voir les notifications envoyer par l'administrateur

The screenshot shows the 'Notifications' interface for a teacher. The sidebar on the left includes options like 'Accueil', 'Profil', 'Ajouter Resultat', 'Modifier Resultat', 'Presence', 'Mettre a Presence', 'Notifications' (highlighted), 'Demander le congés', 'Commentaires', and 'Deconnexion'. The main section is titled 'Notifications' and contains a table with the following data:

ID	Date	Message
1	Nov. 16, 2021, 2:24 p.m.	bonjourmr niang vous allez bien..... vous etes a la maison?

At the bottom of the page, there is a footer that reads: 'Copyright © 2021-2021 SouniouSchool.'

FIGURE 35 – Interface « Notification »

Ici nous avons la partie qui permet au professeur de demander un congé.

The screenshot shows a web application interface for a teacher. On the left is a dark green sidebar with the title 'Professeur' and a user profile for 'Niang, Pape Ibrahima'. The sidebar contains a list of menu items: 'Accueil', 'Profil', 'Ajouter Resultat', 'Modifier Resultat', 'Presence', 'Mettre a Presence', 'Notifications', 'Demander le congés' (highlighted in blue), 'Commentaires', and 'Deconnexion'. The main content area is titled 'Demander un congé' and has a breadcrumb 'Accueil / Demander un congé'. It features a form with a date input field labeled 'Date:' with a placeholder 'jj/mm/aaaa' and a calendar icon, and a text area labeled 'Message:'. At the bottom of the form is a green button labeled 'Appliquer'.

FIGURE 36 – Interface « Demander un congé »

Cette permet d'ajouter un commentaire que l'administrateur va recevoir et envoyer une réponse.

The screenshot shows a web application interface for a teacher. On the left is a dark green sidebar with the title 'Professeur' and a user profile for 'Niang, Pape Ibrahima'. The sidebar contains a list of menu items: 'Accueil', 'Profil', 'Ajouter Resultat', 'Modifier Resultat', 'Presence', 'Mettre a Presence', 'Notifications', 'Demander le congés', 'Commentaires' (highlighted in blue), and 'Deconnexion'. The main content area is titled 'Ajouter commentaires' and has a breadcrumb 'Accueil / Ajouter commentaires'. It features a form with a text area labeled 'Feedback:'. Below the text area is a green button labeled 'Soumettre'. At the bottom of the form is a table with the title 'Ajouter commentaires' and four columns: 'ID', 'Commentaire', 'Reponse', and 'Created At'.

FIGURE 37 – Interface « Ajouter commentaires »

Et maintenant on s'attaque à l'accueil de la partie des apprenants, ou nous avons le total de ses absences, le pourcentage de présence, le pourcentage d'absence et le total des matières que l'apprenant suit, ainsi que le diagramme des présences et des absences et les absences par matière.

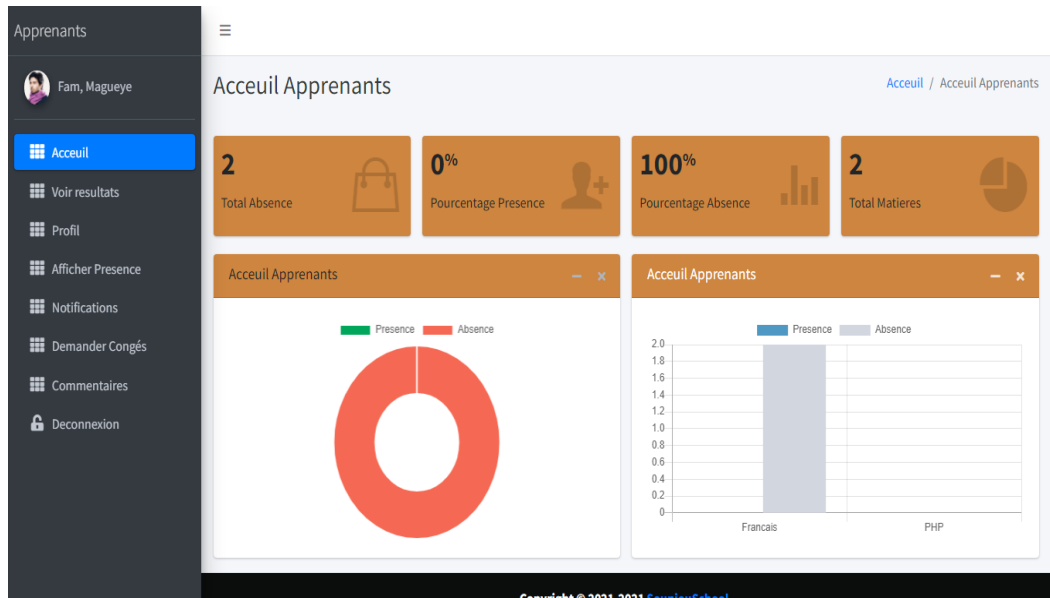


FIGURE 38 – Interface « Accueil élève »

Ici l'apprenant peut voir ses notes en fonction de la matière.

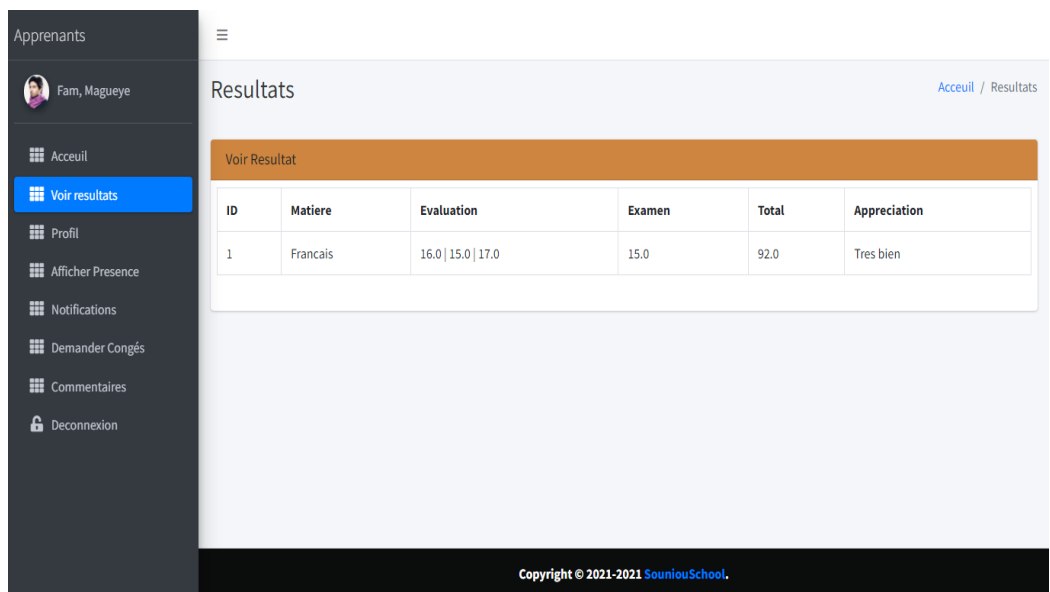


FIGURE 39 – Interface « Notes élèves »



Dans cette partie l'apprenant peut modifier ses identifiants et son image de profil.

FIGURE 40 – Interface « Éditer Profil élève »

Dans cette partie l'apprenant peut voir ses absences en fonction d'une matière et d'un intervalle de date.

FIGURE 41 – Interface « Présence »

Cette partie permet à l'apprenant de demander des congés et d'attendre s'il sera accepté ou refuser.

**Apprenants**

Fam, Magueye

- Accueil
- Voir resultats
- Profil
- Afficher Presence
- Notifications
- Demander Congés**
- Commentaires
- Deconnexion

**Demande**

Date:

Message:

Envoyer

**Demande**

ID	Date	Message	Status
1	2021-11-16	dihfjofojkqsdofiqpfqefqzefioqzefiosifqojcdiosqjciodqsjioej	Rejeté
2	2021-11-16	bonjour monsieur j'ai bapteme a la maison du coup je ne pourrait pas venir a l'ecole .....merci de votre comprehension	Attente

**FIGURE 42 – Interface « Demande autorisation d'absence élève »**

Cette partie permet à l'apprenant d'ajouter des commentaires et d'attendre une éventuelle réponse.

**Apprenants**

Fam, Magueye

- Accueil
- Voir resultats
- Profil
- Afficher Presence
- Notifications
- Demander Congés
- Commentaires**
- Deconnexion

**Commentaire Apprenants**

Feedback:

Soumettre

**Commentaire Apprenants**

ID	Commentaire	Reponse	Date
1	je pense que lecole est mauvais	magueye vous etes serieux ras	Nov. 16, 2021, 1:09 a.m.
2	l'ecole est la meilleure au senegal	Pas de Reponse	Nov. 16, 2021, 12:59 p.m.

**FIGURE 43 – Interface « Commentaire élève »**

## 5. CONCLUSION

L'objectif de notre projet de fin de formation est de concevoir et mettre en œuvre une application web de gestion des élèves et du système. Cette phase pratique de notre formation est dédiée à l'analyse, la conception et la programmation, nous guidant pour répondre aux besoins réels en fonction des objectifs spécifiés et des contraintes rencontrées à partir des problèmes observés.

Pendant le développement, nous avons rencontré quelques difficultés. Django étant un Framework innovant et complet qui utilise les dernières possibilités de Python et adopte l'architecture MVT mais ne l'impose pas. On s'est donc familiarisé avec ces outils informatiques et ces langages.

On a appliqué autant de règles de base que possible nous permettant d'avoir une application puissante. On a utilisé le langage UML pour modéliser le système en différentes étapes, et MySQL comme langage SGBD et Python pour implémenter notre application. Puisque Python est un langage flexible, l'application sera portable et utilisable quelle que soit la plateforme.

La réalisation d'un tel projet nous a donné la possibilité de découvrir de nouvelles approches de développement.

Les résultats de ce projet sont généralement positifs et les principaux objectifs du projet sont atteints. La plupart des travaux qui m'ont été demandés sont terminés. Un petit nombre de modifications ou d'améliorations aux divers programmes qui n'ont pas encore été apportées seront mises en œuvre.

Nous ne prétendons pas être parfaits. C'est pourquoi, nous espérons aller au-delà des imperfections et lacunes que vous remarquerez et rencontrerez, et considérer vos commentaires, critiques et suggestions pour rendre possible l'amélioration du projet.

## 6. PESPECTIVES

Néanmoins les résultats de ce modeste travail constituent les bases d'un travail à poursuivre et à améliorer pour une étude beaucoup plus approfondie qui pourra faire l'objet d'une thèse de doctorat.

Ainsi, les perspectives futures sont dans un premier temps l'Intégration de l'AD pour la création automatiques des comptes utilisateurs.

Puis Imprimer les bulletins et gérer l'emploi du temps

Ensuite Ajouter des interfaces suivantes :

### ***Comptables :***

- ❖ Gérer les paiements et les frais
- ❖ Imprimer les reçus de paiement

### ***bibliothécaires :***

- ❖ Gérer les livres dans la bibliothèque

### ***Parent d'élève :***

- ❖ Voir le profil de l'enseignant
- ❖ Afficher la feuille de notes de son propre enfant (Télécharger/Imprimer le PDF)
- ❖ Voir l'emploi du temps de son propre enfant
- ❖ Voir les paiements de son propre enfant
- ❖ Afficher le tableau d'affichage et les événements de l'école dans le calendrier
- ❖ Gérer son propre profil
- ❖ Faire des contributions

## WEBOGRAPHIE

<https://www.journaldunet.fr/web-tech/>

<https://www.cairn.info/>

<https://developer.mozilla.org/>

[Définition du SQL \(Structured Query Language\) - Actualité Informatique](#)

[\(actualiteinformatique.fr\)](http://actualiteinformatique.fr)

<https://pdfprof.com/>



